

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	19	resource\$4 near6 lock\$5 same peer\$1	US-PGPUB; USPAT; EPO; JPO	OR	OFF	2005/01/21 14:41

US-PAT-NO: 5699500  
DOCUMENT-IDENTIFIER: US 5699500 A  
TITLE: Reliable datagram service provider for fast messaging in a clustered environment  
----- KWIC -----  
Detailed Description Text - DETX (41):  
Shared Memory Header. The shared memory segment is a virtual contiguous section of the address space of each process that connects with the DLM. The memory segment is carved into a header section that stores general information regarding the DLM and its peer nodes, while the body section stores all the data structures that are used to represent resources, locks, hash vectors, process slots etc. The shared memory header contains the following structures:

US-PAT-NO:

5805900

DOCUMENT-IDENTIFIER: US 5805900 A

TITLE: Method and apparatus for serializing resource access

requests in a multisystem complex

----- KWIC -----

Brief Summary Text - BSTX (22):

Each local lock manager uses the global lock manager in the manner described above, distributing the information and management of resource serialization across the sysplex. The structure of the local lock managers is changed from a peer-coupled ring to a star with a local lock manager on each point and the global lock manager in the middle. Each local lock manager is now responsible for managing only its global resource requests. Global resource requests from other systems in the sysplex will no longer be replicated on every sysplex-wide view of resource contention will be kept in the global lock manager, but sysplex-wide global resource queues will not be kept.

US-PAT-NO: 5909540

DOCUMENT-IDENTIFIER: US 5909540 A

TITLE: System and method for providing highly

available data

storage using globally addressable memory

----- KWIC -----

Detailed Description Text - DETX (56):

The shared memory provides the distribution mechanism for resource sharing among peer nodes running the file system 60 software. Each instance of the file system 60 on each network node views the shared memory resources (i.e., pages) as being shared with other local or remote threads. The file system 60 needs a way to implement high level, file system locks to provide consistent resource sharing. Any concurrency control structure can be used to implement locks, such as lock objects or semaphores. In database applications, locking may also be achieved by implementing concurrency control structures associated with database indices or keys. In file system applications access to files or directories may be controlled. Another example of file system locks is Byte Range Locking, which provides the users the ability to coordinate shared access to files. A byte range lock is a lock set on a range of bytes of a coordinated shared access to a file can be accomplished by taking locks on the desired byte ranges. In general, the high level file system lock works in the following fashion: (a) a file system resource is to be shared by each file system 60 instance, and the access to the resource is coordinated by a locking protocol using a lock object data structure that represents the high level lock to coordinate the shared resource, and it is the value of the data structure that represents the current state of the lock; (b) to access the

resource, the instance at each node must be able to look at the state (or value) of the lock data structure, and if it is "free," modify it so that it becomes "busy," but if it is "busy," then it has to wait to become "free," and there could be intermediate states between "free" and "busy" (i.e., more than two lock states), but in any event, in this byte range locking example, a lock is a description of a certain byte range being shared/exclusively locked by some thread of the file system 60, and a conflicting new byte range lock request that falls in or overlaps the already locked byte range will be denied or the requester may block (depending on how the request was made); and (c) access to or modification of the lock data structure by each node's instance needs to be serialized so that it in turn can then be used to coordinate high level resource sharing.

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	1282	cluster\$4 adj2 network	US-PGPUB; USPAT; EPO; DERWENT; IBM_TDB	OR	ON	2005/01/21 13:40
L2	74	1 and node and lock and request and client	US-PGPUB; USPAT; EPO; DERWENT; IBM_TDB	OR	ON	2005/01/21 13:44
L3	26	1 and node and lock adj request and client	US-PGPUB; USPAT; EPO; DERWENT; IBM_TDB	OR	ON	2005/01/21 13:44
L4	26	3 and resource	US-PGPUB; USPAT; EPO; DERWENT; IBM_TDB	OR	ON	2005/01/21 13:44
L5	14	4 and "709"/\$.ccls.	US-PGPUB; USPAT; EPO; DERWENT; IBM_TDB	OR	ON	2005/01/21 13:55
L6	0	4 and peer	US-PGPUB; USPAT; EPO; DERWENT; IBM_TDB	OR	ON	2005/01/21 13:56
L7	26	4 and process	US-PGPUB; USPAT; EPO; DERWENT; IBM_TDB	OR	ON	2005/01/21 14:05
L8	0	7 and cmlock	US-PGPUB; USPAT; EPO; DERWENT; IBM_TDB	OR	ON	2005/01/21 14:05
L9	0	7 and daemon	US-PGPUB; USPAT; EPO; DERWENT; IBM_TDB	OR	ON	2005/01/21 14:10
L10	2	"6795832".pn.	US-PGPUB; USPAT; EPO; DERWENT; IBM_TDB	OR	ON	2005/01/21 14:23

L11	3	"2001005654"	US-PGPUB; USPAT; EPO; DERWENT; IBM_TDB	OR	ON	2005/01/21 14:24
L12	3	"2001005654".pn.	US-PGPUB; USPAT; EPO; DERWENT; IBM_TDB	OR	ON	2005/01/21 14:25
L13	92	chrabaszcz	US-PGPUB; USPAT; EPO; DERWENT; IBM_TDB	OR	ON	2005/01/21 14:26
L14	31	chrabaszcz.in.	US-PGPUB; USPAT; EPO; DERWENT; IBM_TDB	OR	ON	2005/01/21 15:19
L15	498	cluster and node and lock and conflict	US-PGPUB; USPAT; EPO; DERWENT; IBM_TDB	OR	ON	2005/01/21 15:20
L16	384	15 and resource	US-PGPUB; USPAT; EPO; DERWENT; IBM_TDB	OR	ON	2005/01/21 15:20
L17	207	16 and client	US-PGPUB; USPAT; EPO; DERWENT; IBM_TDB	OR	ON	2005/01/21 15:20
L18	25	17 and 4	US-PGPUB; USPAT; EPO; DERWENT; IBM_TDB	OR	ON	2005/01/21 15:26
L19	6912	resource adj allocation	US-PGPUB; USPAT; EPO; DERWENT; IBM_TDB	OR	ON	2005/01/21 15:26
L20	53	15 and 19	US-PGPUB; USPAT; EPO; DERWENT; IBM_TDB	OR	ON	2005/01/21 15:34

L21	1	20 and 18	US-PGPUB; USPAT; EPO; DERWENT; IBM_TDB	OR	ON	2005/01/21 15:26
L22	0	15 and 19daemon	US-PGPUB; USPAT; EPO; DERWENT; IBM_TDB	OR	ON	2005/01/21 15:34
L23	3972	daemon	US-PGPUB; USPAT; EPO; DERWENT; IBM_TDB	OR	ON	2005/01/21 15:34
L24	14	20 and 23	US-PGPUB; USPAT; EPO; DERWENT; IBM_TDB	OR	ON	2005/01/21 15:36
L25	14	24 and conflicts	US-PGPUB; USPAT; EPO; DERWENT; IBM_TDB	OR	ON	2005/01/21 15:37
L26	10	25 and peer	US-PGPUB; USPAT; EPO; DERWENT; IBM_TDB	OR	ON	2005/01/21 15:37



